Claims

WHAT IS CLAIMED IS:

1. A method of comparing first and second databases that are each comprised of a plurality of entities having one or more characteristics, said entities being grouped into a plurality of data classes in each said database each representative of a particular entity type, the method comprising:
(i) for each said data class of said first and second databases, compiling a list representative of the entities occurring within that class and any attributes for each said entity;
(ii) identifying and comparing corresponding data classes for each of said first and second databases; and
(iii) identifying on the basis of said comparison differences between corresponding entities of said corresponding data classes.

2. A method according to Claim 1, wherein said first and second databases are relational databases.

3. A method according to Claim 2, wherein each structure of said first and second databases may be described by means of a schema.

4. A method according to Claim 3, wherein said first and second databases conform to substantially the same schema.

5. A method according to Claim 3, wherein said first and second databases conform to different schemas.

6. A method according to Claim 3, wherein said first and second schema conform to different versions of the same schema.

7. A method according to Claim 1, wherein said first and second databases are capable of being represented in XML file format.

8. A method according to Claim 1, wherein said identifying step additionally comprises identifying entities that have been added to one or other of said first and second databases.

9. A method according to Claim 1, wherein said identifying step additionally comprises identifying entities that have been deleted from one or other of said first and second databases.

10. A method according to Claim 1, wherein said differences between entities include one or more of:
(a) the absence of a value of a said entity characteristic in one entity of corresponding entities in said first and second databases;

(b) a modification of a value of a said entity characteristic in one entity of corresponding entities in said first and second databases.

11. A method according to Claim 1, comprising the additional step - in circumstances where said databases conform to an identical schema or different versions of a schema - of automatically detecting the identity of said schema.

12. A method according to Claim 1, wherein said compiling step includes the steps of reviewing said databases for characters with an encoding incompatible with the parser, and translating any such character to an equivalent character with compatible encoding.

13. A method according to Claim 1, wherein said compiling step includes the step of parsing each of said first and second databases.

14. A method according to Claim 13, wherein said parsing step comprises instantiating an object class of appropriate type for each entity.

15. A method according to Claim 14, wherein an object is instantiated for each entity of said first and second databases.

16. A method according to Claim 15, wherein the list compiled for said first database includes entities of said first database, and the list compiled for said second database includes entities of said second database.

17. A method according to Claim 15, wherein said identifying and comparing step includes a step of instantiating a difference object for each entity listed in one of the lists for said first and second databases.

18. A method according to Claim 17, further comprising the step of populating each said difference object with a reference to the corresponding entity object in said one list.

19. A method according to Claim 18, wherein for speed and efficiency of later searching, a reference to each said entity object is created in a hash keyed by object identity.

20. A method according to Claim 18, comprising the step of computing a lineage (recursive parenthood) for each said entity object in said one list.

21. A method according to Claim 20, comprising storing said computed lineages in a lineage list.

22. A method according to Claim 21, wherein for speed and efficiency of later searching, a reference to said difference object is saved in a hash keyed by lineage.

23. A method according to Claim 22, further comprising computing a lineage (recursive parenthood) for each entity of said other of the lists for said first and second databases.

24. A method according to Claim 23, wherein said comparing step includes comparing the computed lineages for each entity of said other of the lists with the lineages computed for the entities in said one list.

25. A method according to Claim 24, wherein in the event of a match between a computed lineage for an entity of said other of the lists and a lineage computed for an entity in said one list, a potential entity match is determined to have occurred.

26. A method according to Claim 25, wherein in the event of a potential entity match occurring, a unique identifier for each potentially matching entity is retrieved and compared, and an actual match is determined to have occurred if the unique identifiers should be determined to match.

27. A method according to Claim 26, wherein in the event of an actual match occurring, the difference object corresponding to the matched entity referenced in said one list is populated with a reference to the matched entity referenced in said other list.

28. A method according to Claim 26, wherein in the event of an actual match not occurring, a difference object is instantiated for said unmatched entity in said other list.

29. A method according to Claim 28, wherein said difference object is populated with a reference to said unmatched entity from said other list.

30. A method according to Claim 29, wherein for speed and efficiency of later searching, a reference to said difference object is saved in said hash keyed by lineage.

31. A method according to Claim 29, wherein said identifying step includes inspecting each said difference object for definition of the entity object or objects referenced therein.

32. A method according to Claim 31, further comprising the step of determining an entity addition or deletion, as appropriate, to have occurred on identification of a said difference object with a single reference to an entity object.

33. A method according to Claim 31, wherein said identifying step includes in circumstances where a said difference object includes a reference to two entity objects, retrieving and comparing each characteristic of said entity objects.

34. A method according to Claim 33, further comprising the step of reporting, as differences, modifications, additions and/or deletions of characteristics of said two entities.

35. A method according to Claim 34, further comprising reporting, in the case of a modification, the characteristic of each of said two entities.

36. A method according to Claim 1, further comprising the step of presenting options to a user concerning the manner in which the first and second databases are to be compared.

37. A method according to Claim 36, wherein said user is capable of choosing one of a number of identifiers for subsequent use in entity matching.

38. A method according to Claim 37, wherein said identifiers comprise for each said entity one or more of: a name, a serial number by entity type (table key), or a globally unique identifier (GUID).

39. A method according to Claim 38, wherein said options are provided to said user by means of a graphical user interface or a configurable XML file.

40. A method according to Claim 39, wherein said user is also capable of selecting which entity types and characteristics are compared.

41. A method according to Claim 39, wherein said user is capable of limiting comparison to specified parts of each of said first and second databases.

42. A method according to Claim 39, wherein a user's option selections are accepted via a parsable configuration file in XML format.

43. A method according to Claim 42, wherein GUI-modified options can be stored in a parsable configuration file in XML format.

44. A method according to Claim 43, wherein said options are instantiated as configuration objects.

45. A method according to Claim 1, further comprising the step of automatically generating a report of any identified differences.

46. A method according to Claim 45, wherein said report can be generated in one or more of a plurality of file formats, each being selectable by a user.

47. A method according to Claim 26, wherein said comparing step employs fuzzy matching to permit automated pairing of entities of identical type but of only similar attribute values.

48. A method according to Claim 47, wherein the extent of fuzzy matching permitted is user selectable.

49. A computer program comprising one or more software portions which, when executed in an execution environment, are operable to perform the method steps set out in Claim 1.

50. A storage medium encoded with machine readable computer program code for comparing first and second databases that are each comprised of a plurality of entities having one or more characteristics, said entities being grouped into a plurality of data classes in each said database each representative of a particular entity type; wherein, when the computer program code is executed by a computer, the computer performs the steps of:

i) for each said data class of said first and second databases, compiling a list representative of the entities occurring within that class and the attributes

for each said entity;

(ii) identifying and comparing corresponding data classes for each of said first and second databases, and

(iii) identifying on the basis of said comparison differences between corresponding entities of said corresponding data classes.

51. A comparator for comparing first and second databases that are each comprised of a plurality of entities having one or more characteristics, said entities being grouped into a plurality of data classes in each said database each representative of a particular entity type, the comparator comprising:

(i) a first module which is operable, for each said data class of said first and second databases, to compile a list representative of the entities occurring within that class and the attributes for each said entity;

(ii) a second module that is operable to identify and compare corresponding data classes for each of said first and second databases, and

(iii) a third module which is operable to identify, on the basis of said comparison, differences between corresponding entities of said corresponding data classes.

52. A comparator according to Claim 51, implemented in hardware.

53. A comparator according to Claim 51, wherein said hardware comprises an application specific integrated circuit.

54. A comparator according to Claim 51, implemented in software.

55. A storage medium encoded with machine readable computer program code for comparing a first older database and a second newer database each conforming to different versions of a database structure schema, said first and second databases each being comprised of a plurality of entities having one or more characteristics; wherein, when the computer program code is executed by a computer, the computer performs the steps of:

(i) automatically generating object-orientated data structures from schema objects relating to said schema versions;

(ii) parsing said first and second databases;

(iii) populating instances of said data structures with instances of the appropriate entities for each said database;

(iv) generating lists of entities for each entity type instanced in the said data structures for each said database;

(v) pairing corresponding entities in said lists;

(vi) populating a third instance of said data structures with instances of appropriate difference entity types;

(vii) storing in said difference entity types and hence in said third data structure the results of said pairing;

(viii) scanning said results of said pairing to determine which entities in said first database have been deleted, which entities in said second database have been inserted, and which entities are common to said first and second databases;

(ix) comparing characteristics of said common entities to

determine which of the said common entities have been modified and which remain unmodified; and

(x) reporting said deletions, insertions, modifications, non-modifications and results of said characteristic comparisons.

56. A method for comparing a first older database and a second newer database each conforming to different versions of a database structure schema, said first and second databases each being comprised of a plurality of entities having one or more characteristics, the method comprising the steps of:

(i) automatically generating object-orientated data structures from schema objects relating to said schema versions;

(ii) parsing said first and second databases;

(iii) populating instances of said data structures with instances of the appropriate entities for each said database;

(iv) generating lists of entities for each entity type instanced in the said data structures for each said database;

(v) pairing corresponding entities in said lists;

(vi) populating a third instance of said data structures with instances of appropriate difference entity types;

(vii) storing in said difference entity types and hence in said third data structure the results of said pairing;

(viii) scanning said results of said pairing to determine which entities in said first database have been deleted, which entities in said second database have been inserted, and which entities are common to said first and second databases;

(ix) comparing characteristics of said common entities to determine which of the said common entities have been modified and which remain unmodified; and

(x) reporting said deletions, insertions, modifications, non-modifications and results of said characteristic comparisons.